

**Original citation:**

Ramis Ferrer, Borja, Ahmad, Bilal, Vera, Daniel, Lobov, Andrei, Harrison, Robert and Martínez Lastra, José Luis. (2016) Product, process and resource model coupling for knowledge-driven assembly automation. *Automatisierungstechnik*, 64 (3). pp. 231-243.

**Permanent WRAP URL:**

<http://wrap.warwick.ac.uk/78817>

**Copyright and reuse:**

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

**Publisher's statement:**

© De Gruyter 2016

<http://dx.doi.org/10.1515/auto-2015-0073>

**A note on versions:**

The version presented in WRAP is the published version or, version of record, and may be cited as it appears here.

For more information, please contact the WRAP Team at: [wrap@warwick.ac.uk](mailto:wrap@warwick.ac.uk)

## Anwendungen

Borja Ramis Ferrer\*, Bilal Ahmad, Daniel Vera, Andrei Lobov, Robert Harrison, and José Luis Martínez Lastra

# Product, process and resource model coupling for knowledge-driven assembly automation

Kopplung von Produkt-, Prozess- und Ressourcenmodell für die wissensgetriebene Montageautomation

DOI 10.1515/auto-2015-0073

Received October 15, 2015; accepted December 30, 2015

**Abstract:** Accommodating frequent product changes in a short period of time is a challenging task due to limitations of the contemporary engineering approach to design, build and reconfigure automation systems. In particular, the growing quantity and diversity of manufacturing information, and the increasing need to exchange and reuse this information in an efficient way has become a bottleneck. To improve the engineering process, digital manufacturing and Product, Process and Resource (PPR) modelling are considered very promising to compress development time and engineering cost by enabling efficient design and reconfiguration of manufacturing resources. However, due to ineffective coupling of PPR data, design and reconfiguration of assembly systems are still challenging tasks due to the dependency on the knowledge and experience of engineers. This paper presents an approach for data models integration that can be employed for coupling the PPR domain models for matching the requirements of products for assembly automation. The approach presented in this paper can be used effectively to link data models from various engineering domains and engineering tools. For proof of concept, an example implementation of the approach for modelling and integration of PPR for a Festo test rig is presented as a case study.

**Keywords:** Knowledge driven systems, model coupling, ontology matching, assembly automation.

**Zusammenfassung:** Die Berücksichtigung häufiger Produktänderungen innerhalb kurzer Zeiträume ist eine Herausforderung für Ingenieure aufgrund der Limitierungen aktueller Ansätze zum Entwurf, Aufbau und Rekonfiguration von Automatisierungssystemen. Insbesondere die steigende Menge und Vielfalt an Informationen aus der Produktion, die auf effiziente Art und Weise ausgetauscht und wiederverwendet werden sollen, führt zu einem Engpass. Zwei vielversprechende Ansätze um Entwicklungskosten und Entwicklungszeit zu reduzieren und gleichzeitig eine effiziente Entwicklung sowie die Rekonfiguration des Montagesystems zu ermöglichen, sind die digitale Fabrik und kombinierte Produkt-, Prozess- und Ressourcenmodelle (PPR). Aufgrund der bisher ineffizienten Verknüpfung der PPR-Daten ist die Entwicklung und die Rekonfiguration von Montageanlagen noch immer eine sehr anspruchsvolle Aufgabe aufgrund der starken Abhängigkeit vom Fachwissen und der Erfahrung der Ingenieure. Dieser Artikel präsentiert einen Ansatz zur Integration von Datenmodellen, der zur Kopplung der PPR-Domänenmodelle eingesetzt werden kann, um die Anforderungen der Produkte an die Montageautomatisierung zu erfüllen. Die hier vorgestellte Methode dient dazu, Datenmodelle und ihre Werkzeuge aus verschiedenen Ingenieursdisziplinen zu koppeln. Zur Evaluation wird eine beispielhafte Implementierung des Modell und Integrationsansatzes die Fallstudie eines FESTO-Prüfstandes präsentiert.

**Schlüsselwörter:** wissensbasierte Systeme, Koppelmodell, ontologische Übereinstimmung, Montageautomatisierung, Produktionsautomatisierung.

\*Corresponding author: Borja Ramis Ferrer, Tampere University of Technology – FAST-Lab. P.O. 600, FI-33101, Tampere, Finland, e-mail: borja.ramisferrer@tut.fi

Andrei Lobov, José Luis Martínez Lastra: Tampere University of Technology – FAST-Lab. P.O. 600, FI-33101, Tampere, Finland

Bilal Ahmad, Daniel Vera, Robert Harrison: WMG, University of Warwick, Coventry, CV4 7AL, United Kingdom

## 1 Introduction

The increasing market turbulence and the rise of new product technologies represent challenges and force industry to manufacture new product variants and adjust

production volume constantly. For example, the automotive industry is expected to accommodate changes frequently due to increased environmental concerns, technological advancements, changes in the market requirements etc. As a result, product assembly is becoming a challenging task. Industries have to frequently change manufacturing process plans and subsequently reconfigure/build assembly lines to manufacture new products [1].

Assembly systems within the automotive manufacturing sector are typically based on commonality approaches. Around 80% of existing off-the-shelf manufacturing resources remain unchanged or slightly modified to accommodate a new version of a product. However, due to the use of ad hoc engineering methods the knowledge utilised from previous similar projects is only a small percentage of the available knowledge [2]. It is reported in [3] that companies are often unaware of the extent of in-house knowledge and therefore spend significant time searching for relevant information from previous similar projects.

To accommodate product changes, the required manufacturing process and associated resource constraints are typically checked manually due to the lack of tools that can support analysis of how the product changes will affect manufacturing operations. As a result, feasibility studies associated with the manufacturing of new products on existing assembly lines, design of new process plans and required reconfiguration of manufacturing resources remain subjective. Such ad hoc approaches rely heavily on the knowledge and experience of engineers that result in prolonged lead-times and increased engineering costs [4].

To assist in the design, planning and reconfiguration of assembly systems, a number of digital modelling and simulation tools have been developed. These tools provide a number of benefits such as visualisation, verification and optimisation of the manufacturing systems before the physical build [5, 6]. However, manufacturing process and resource modelling is still a challenging and complex task. This is mainly due to the gaps that exist in the engineering process due to the reliance on ad hoc mechanisms for knowledge sharing, integration and reusability. Typically, product, process, and resource information and data sets exist within a given organisation, but they are not effectively coupled [7], which represents a big challenge in this field.

One promising approach to address this issue is the effective linking of the digital descriptions of the product attributes and requirements, to the characteristics of the manufacturing resources. The resulting knowledge-base derived from such integrated digital description can potentially increase the efficiency of the engineering pro-

cess. The focus of this paper is to improve PPR modelling of assembly automation systems by integrating virtual engineering tools with an ontology-based knowledge modelling system. To realise such modelling approach, a generic concept is presented that couples data models of PPR. The matching of different ontology domain models in a knowledge-driven system can potentially provide reusable knowledge-based PPR description to interconnect product attributes with related manufacturing resources.

In this article, the authors present a method for coupling PPR data models that can be achieved within the implementation of the rule-based approaches described in [4, 8] for mapping data of different concepts. In other words, this research work presents an application of previous work for multiple domain ontologies matching. It should be noted that previous work focused directly on mapping data in a unique and artificial model. In fact in a real scenario, these models can be generated from the domain specific engineering tools by corresponding domain experts. For this, the use of standard terminology and taxonomies is a promising approach, where standardisation can be driven by relevant communities holding the expert knowledge on a particular domain. Based on query rules, the presented methodology allows mapping between product, process and resource ontology models that will become modules of the PPR model. The ontology-based representation of engineering knowledge permits the dynamic modelling and mapping of products to required assembly processes and resources. Moreover, the use of standard languages for designing ontological models means that data can be accessed by third party applications, allowing retrieval and updating of the data model and thus makes the proposed approach extensible and scalable.

The remainder of this paper is organised as follows. Section 2 presents the background of the research work. Section 3 describes the approach for PPR model coupling within semantic descriptions. Section 4 presents an implementation of the approach with a case study based on a Festo test rig. Finally, Section 5 concludes the research work and discusses further tasks.

## 2 Background

The recent developments in manufacturing system engineering methods and software tools have focused on extending the data set based on which the design of production system is conducted; PLM (Product Lifecycle Management) and PDM (Product Data Management) have been

integrated with MPM (Manufacturing Processes Management) tools in order to achieve better integration between product design and manufacturing system design processes [9]. Paradigms such as design for assembly, fabrication, manufacture, etc. focus on integrating critical design parameters and constraints, early in the product design phase, in order to achieve higher level of concurrency and avoid deviations between product and production system design processes [10].

Within PLM systems, the concept of PPR extends the above mentioned concepts by integrating the notion of standardised manufacturing processes and resources, and the relation that exists between processes and resources [11]. In the domain of manufacturing, resources typically represent standardised modular machine units defined at various level of granularity (e.g. component, station, cell, zone, etc.). Each resource supports a specific process and may consist of sub-processes and sub-resources (e.g. a station-level transport system consisting of sub process and resources such as pre-stop, machine-stop and clamping operations).

A typical PPR-based engineering workflow consists of initiating the product design while the manufacturing process and resources required to manufacture this product are concurrently derived from the PPR information mapping. Existing PPR-based systems (e.g. Dassault Enovia, Siemens TeamCenter) typically implement PPR capabilities through the storage and integration (i.e. cross-referencing) of PPR data models. The increasing availability and usage of 3D based virtual modelling and simulation tools within PPR tool chain have significantly contributed to i) extending the data set integrated into PPR systems (mechanical/3D data, station/line layout, PLC control data in the form of re-usable Function Blocks, etc.) and ii) improving design support and validation functions through the use of dynamic 3D-based simulation environment that provide an intuitive view of the complete data set and therefore of the final systems design.

PPR systems rely heavily on the use of relational database systems to store information and to define relations between PPR data sets; relationships between data sets are based on physical or logical constraints and relations (e.g. processing time, product size/resource capacity, etc.), or on specific relations derived from experience collected during past projects (e.g. specific product variant/resource mapping). However, today's deployment of advanced ICT systems and the exponentially increasing quantity of digital data generated, represents a real challenge in not only managing the data (i.e. storage, categorisation, storage), but also in defining and maintaining the

relation between various engineering data sets and models.

The emergence of cyber-physical systems (CPS) integration and the expected industry evolution (Industry 4.0), Knowledge Representation (KR) is a promising solution to manage the referencing and integration of large and expending data sets. This permits the reduction of data processing overload and, at the same time, the support of advanced reasoning and design capabilities provided by knowledge-based systems.

KR is the field of study concerned with describing facts in a human and machine-readable format, which makes machines capable of automating processes with the usage of semantic descriptions. Aside from formally defining KR, [12] presents several formalisms that can be used for implementing KR. During the last decade, the industrial automation research community has tended to implement ontologies for formal description of manufacturing systems [13]. However, it should be noted that Linked Data [14] is considered as a simple method for managing interrelated data in knowledge-based systems but with several challenges for reasoning [15]. On the other hand, other KR formalisms such as semantic rules or frames can be used for representing knowledge.

Ontologies contain formal descriptions, which can be queried and even reasoned. An ontological model is employed as a data storage that contains the actual status of systems, which are semantically described through axioms and relationships between instances of real world objects. In addition, the design of models based on standards permits different designers to convey and describe knowledge with same terminologies and taxonomies. Although there are numerous semantic languages that can be used for modelling manufacturing systems, Resource Definition Framework (RDF)<sup>1</sup> based languages are the dominant ones for model implementation in factory automation [16]. RDF is an XML-based language that belongs to the W3C standard recommendations.

RDF-based models, or RDF graphs, are sets of RDF triples that permit the semantic description of any domain. Syntactically, triples are structured within a relation of three terms: subject, predicate and object. Furthermore, the Ontology Web Language (OWL) [17] is an RDF-based language that extends RDF within the declaration of class descriptions (e.g. enumeration, property restriction, cardinality constraints) and axioms allowing the enrichment of model descriptions. Hence, ontologies not only link data but also add semantics (or meaning) to model resources.

---

<sup>1</sup> <http://www.w3.org/RDF/>

The information can be retrieved from models and modified with the use of standard query languages. This feature allows the data of graph accessible by other applications. In fact, any RDF-based graph datasets can be queried within SPARQL [18] and updated via its extension SPARQL Update (SPARUL). SPARQL permits the extraction of model data in the form of result sets or RDF graphs. SPARUL is an extension of SPARQL that includes a set of operations for updating, creating and removing data from RDF-based models.

The utilisation of semantic descriptions in factory automation domain permits the development of knowledge-driven solutions that are capable of managing and even orchestrating the execution of operations in contemporary production lines. For instance, [19] present useful applications of KR in the factory automation domain that allows scalability and reconfigurability of intelligent industrial automation systems. Implicit data of ontologies can be inferred by semantic reasoning engines to evaluate model descriptions and define new facts, which are not beforehand explicitly described [12]. This is a powerful characteristic of ontologies since the evaluation of models, inclusion at runtime, can result in new classifications or assertions of data. Furthermore, reasoning allows restructuring of models with statements which are not visible or deducible in the design phase. In addition, description of semantic rules using Semantic Web Rules Language (SWRL) [20] can be employed for mapping data from different domains [4, 8] to achieve model coupling.

Due to the number of domains involved in the engineering of manufacturing systems, the data generated during the design phase is heterogeneous and is defined in different domain models. Therefore, effective model coupling and ontology matching becomes an important research focus. [21] describes a conceptual methodology for merging heterogeneous data from upper levels. On the other hand, an example for lower level data coupling is shown in [22], which shows an implementation of algorithms for generating unique models as aggregation of sub-models. The evaluation of mentioned studies and previous work done in [4, 8] motivated and inspired the presented approach in this article.

### 3 Approach

The extensive use of virtual engineering tools in assembly automation for manufacturing process planning, optimisation and validation is promising. Virtual engineering tools have shifted work activity from serial to parallel and advanced information and communication infrastructure

has replaced paper-based processes. Despite this, launching a new product variant in the automotive industry remains a challenge. The selection of required manufacturing resources heavily rely on knowledge and experience of engineers. Although the PPR information exist separately, the lack of mappings between those data sets makes it difficult to identify whether the manufacturing process could be executed using available resources. To address this, an approach for automating the mappings of products, processes and resources using semantic descriptions for assembly automation systems is presented in this section.

The use of modular ontologies (each module belongs to a different domain) and standard terminology and taxonomies is a promising approach to achieving such PPR ontology coupling for assembly systems. Integration of modules in a unique representation creates a common source from which system information can be accessed. However, the main problem with using modular ontology is to define a manner of mapping data to be carried out by designers or with automatic approaches. In fact, the use of common terminology and taxonomies already presented in standards that are being implemented in the industry might avoid data naming convention issues. This requires employment of several standards for covering data managed in all levels of automation systems. For instance, the ISA 88 (used for defining the control batch processes) and the ISA 95 (used for describing the interface between enterprise) can be integrated [23] and used for describing related taxonomy in the PPR domain ontologies.

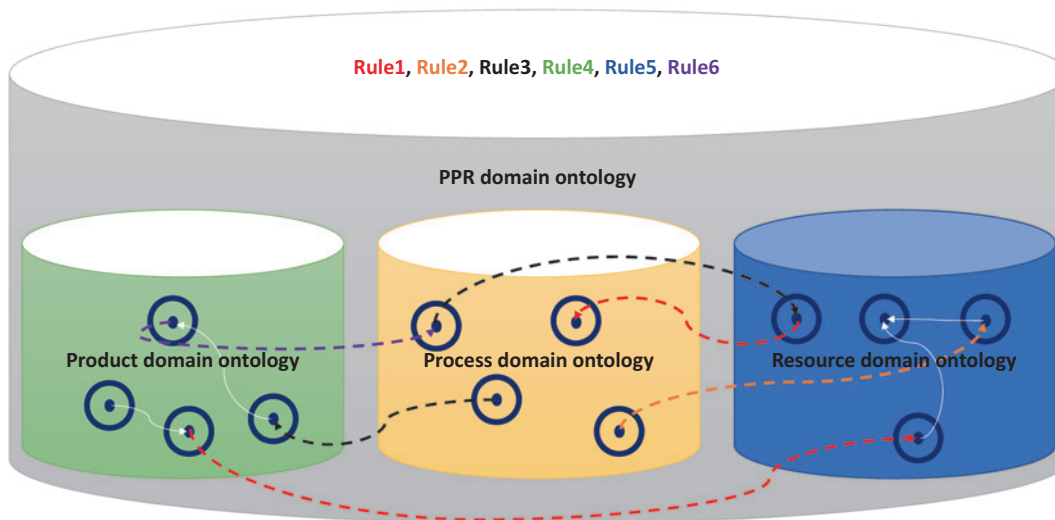
On the other hand, one of the problems that exists nowadays is the lack of a large infrastructure for sharing ontologies that could be reused for describing different industry domains. In fact, the common practice is to develop in-house models that are frequently described in other formats than ontologies e.g. XML, UML models, MS Office documents etc. Nevertheless, models can be transformed to ontologies automatically using existing approaches [24, 25] besides creating them manually and from the scratch. Moreover, APIs as e.g. Apache POI<sup>2</sup> permits the manipulation of MS documents with Java so they can be transformed e.g. to XML objects.

The focus of this paper is working with different domain ontologies and demonstrate how they can be coupled by mapping ontology elements, achieved by importing ontologies and combining them through a rule-based approach. The implementation of this approach allows merging modular ontologies and therefore create a link be-

---

<sup>2</sup> <https://poi.apache.org/>





**Figure 1:** Representation of the resulting PPR domain ontology through different domain model coupling.

tween concepts and data covered by different domains that are frequently described with non-connected standards.

As this approach is intended for the case of having data in different ontologies, the mapping of data is in fact an interconnection between different domain models to address the model coupling of Product, Process and Resource domain ontologies. Figure 1 depicts that how instances belonging to different domain ontologies are matched with rules.

Taking into account the design of modular ontologies [26], different models can be merged to form a unique higher level ontology. This is achieved by using existing ontology editors (e.g. Protégé<sup>3</sup> editor) that form a graph dataset containing all graphs of imported ontologies.

Nevertheless, merging ontologies does not result in automatic creation of link between data belonging to different domains because tools (or an import action) only create a model containing all instances of imported models. Thus, data coupling requires an extra process step. As shown in Figure 1, the model coupling is achieved through the definition of rules that link different data domains. A semantic reasoner that evaluates the model and stated rules infers the interrelation between different data models. It should be noted that creating the PPR ontology and data mapping semantic rules does not remove existing links of the individual domain ontologies, provided they do not create conflict with new rules. The important characteristic of achieving model coupling with the use of rules

and a reasoning engine is that the reasoner automatically validates the consistency of the model.

The approach for domain model coupling of PPR data used in this research consist of following four main steps:

1. Importing domain models in a unique ontology
2. Define a set of object properties that relate different domain concepts
3. Define a set of rules that can be understood by reasoning engines to infer model coupling
4. Automatically infer links between instances and validate model consistency

In the first step, a first graph data set, known as the PPR ontology, is created as an aggregation of all imported models. This new model contains all the elements of each ontology, and have its own Internationalised Resource Identifier (IRI). In the second step, additional object properties that interrelate different domain concepts are defined. These new properties are added in the PPR ontology IRI. Once object properties are added, a set of rules are defined in the next step. Finally, the semantic reasoning engines infer implicit data based on the rules defined in the previous step. The implementation of each step is further described in the case study.

The proposed approach presents a method for model coupling that can be used for e.g. mapping the required data needed for manufacturing products in assembly lines. The data mapping is achieved with an ontological model that allows not only the description of assembly lines, but also the representation of the processes that components can perform. The ontology that is formed by the three PPR domains and main classes used for the im-

<sup>3</sup> <http://protege.stanford.edu/>

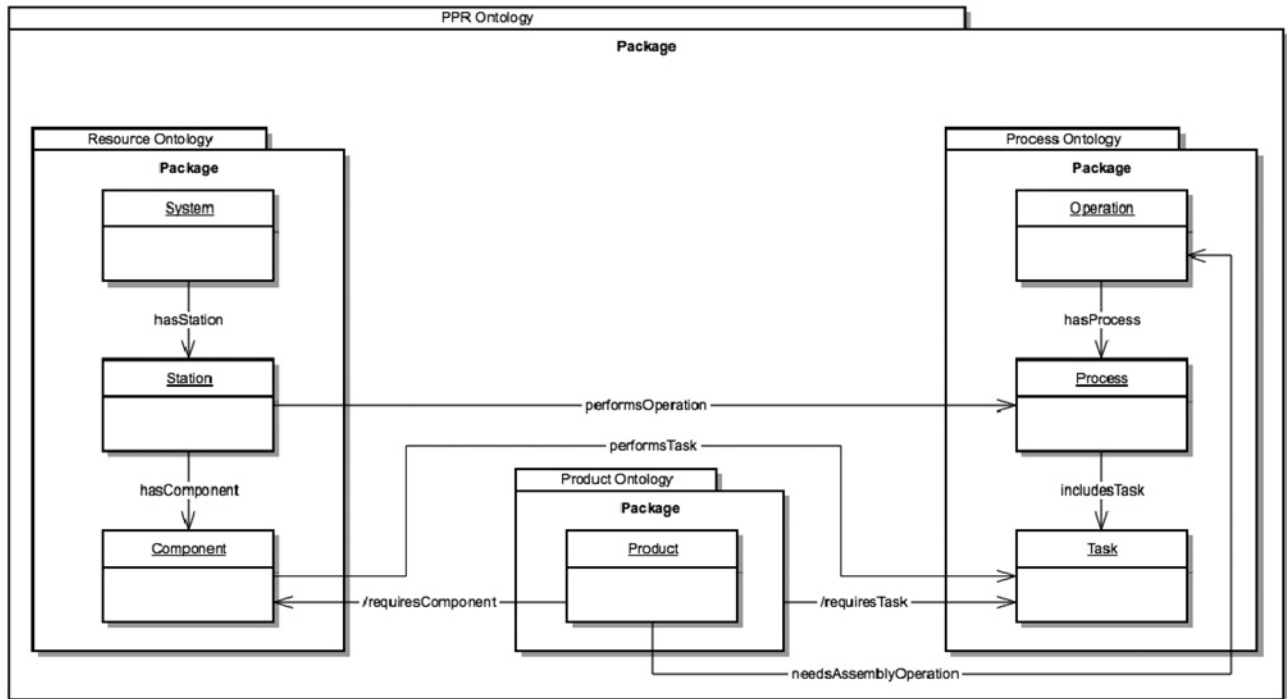


Figure 2: Processing Station ontology model.

plementation of the four step approach are shown in Figure 2 with a UML class diagram. The diagram depicts the hierarchical distribution of packages (ontologies) and included classes, which permit the description of processes and resources of assembly lines for manufacturing products.

Figure 2 depicts the resultant model. This diagram is used as a reference for required descriptions of the research work implementation. Although domain models are usually larger than the ones presented, the described classes are sufficient for demonstrating how the approach can be implemented. It should be noted that the model is reused from the research work presented in [4]. However, different concepts are now separated in different ontologies which are treated as modules (or higher classes) of the PPR ontology model.

## 4 Case study

This section describes the implementation details of the approach presented in previous section with the help of a case study. The case study is based on a station of a Festo Modular Production System, which presents a small-scale realistic industrial test bench widely used for teaching and research purpose. Figure 3 shows the Festo test rig, composed of four stations: *Distributing*, *Buffering*, *Processing*

and *Handling* stations. The *Processing Station* is selected for this use case.

### 4.1 Creating the PPR ontology model

The first step of the approach is accomplished by importing all the ontology models in a unique model. This is carried out using an ontology editor that permits importing models and saving the resulting ontology in a file. Protégé is an ontology editor that is used to implement the presented approach. Figure 4 shows the *Active Ontology* tab from the Protégé interface, in which the imported ontologies information is shown.

As it can be seen in Figure 4, imported ontologies retain the IRI which allow the differentiation between elements that belong to different domain models (even if the name of the instances is same). Therefore, the difference in IRI forms a basis for querying and defining rules in the model.

After the importation of the process ontology, the ontology contains three classes that allow the process related descriptions: *Operation*, *Process* and *Task*. *Operation* class is composed of operations that are performed in system stations. *Process* class is composed of processes linked to operations and *Task* is composed of tasks of processes.

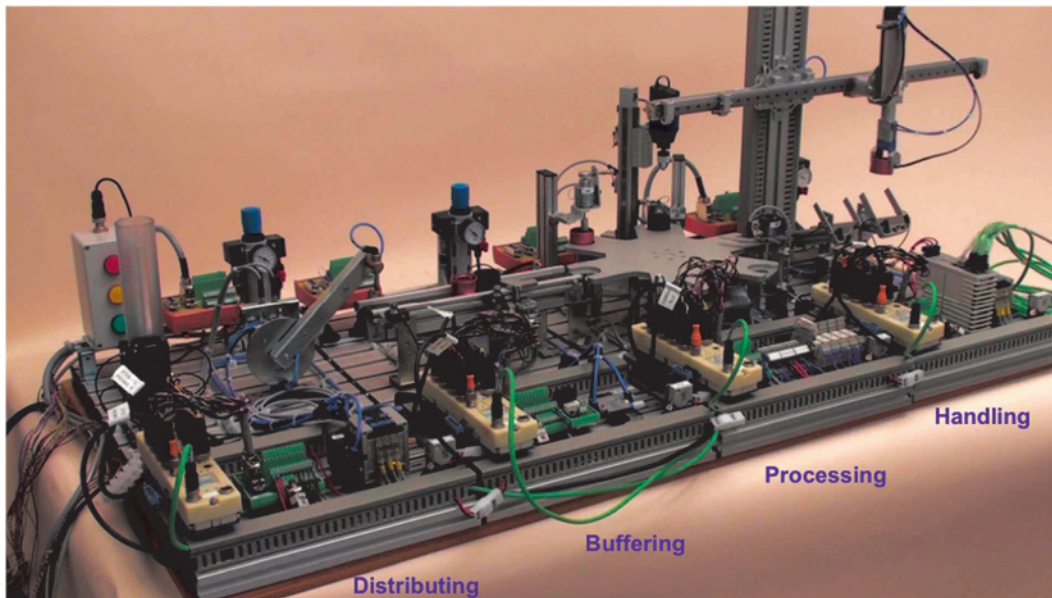


Figure 3: Festo test rig.

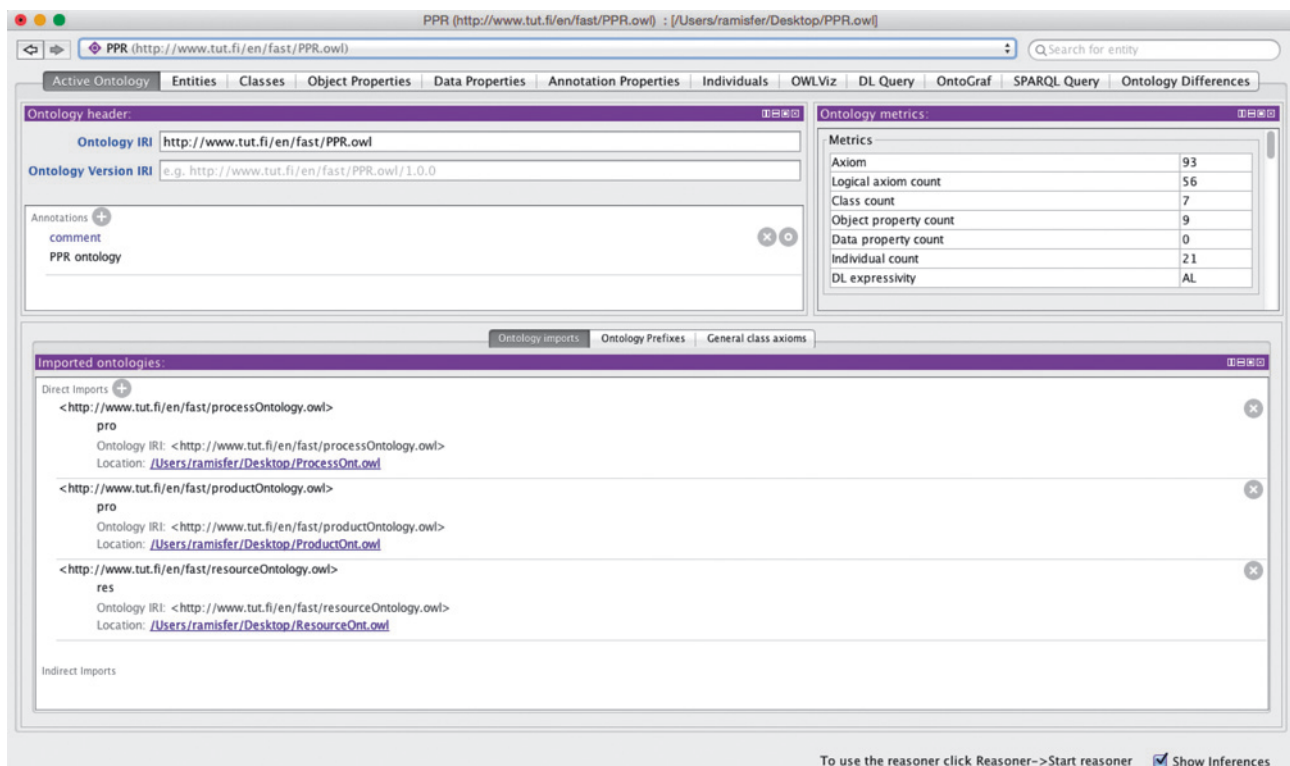


Figure 4: Imported ontologies in the PPR ontology.

A process is defined as a set of tasks performed in a certain sequence.

With the import of product ontology, *Product* class is imported and included in the PPR ontology which is used for collecting types of products that are manufactured in the assembly line. All required relations for PPR mappings

are shown in the class diagram of Figure 2, which is described in section 4.2. These relations are implemented as ontology object properties because they are used as a relationship between class instances.

Finally, after importing the resource ontology, three classes are included in the PPR ontology that allow the



**Table 1:** Instances Property Assertions included in the PPR ontology model.

Instance ( <i>type</i> )	Object property	Instance ( <i>type</i> )
product_1 ( <i>Product</i> )	needsAssemblyOperation	op_01 ( <i>Operation</i> )
product_2 ( <i>Product</i> )		op_02 ( <i>Operation</i> )
festoSystem ( <i>System</i> )	hasStation	processingStation ( <i>Station</i> )
processingStation ( <i>Station</i> )	hasComponent	rotaryInTableModule ( <i>Component</i> )
		drillingModule ( <i>Component</i> )
		testingModule ( <i>Component</i> )
		clampingModule ( <i>Component</i> )
		sortingGateModule ( <i>Component</i> )
rotaryInTableModule ( <i>Component</i> )	performsTask	advance_position ( <i>Task</i> )
drillingModule ( <i>Component</i> )		drill ( <i>Task</i> )
testingModule ( <i>Component</i> )		move_up ( <i>Task</i> )
		move_down ( <i>Task</i> )
clampingModule ( <i>Component</i> )		push_clamp ( <i>Task</i> )
		release_clamp ( <i>Task</i> )
sortingGateModule ( <i>Component</i> )		push_sort ( <i>Task</i> )
		release_sort ( <i>Task</i> )
op_01 ( <i>Operation</i> )	hasProcess	process_1 ( <i>Process</i> )
op_02 ( <i>Operation</i> )		process_2 ( <i>Process</i> )
process_1 ( <i>Process</i> )	includesTask	advance_position ( <i>Task</i> )
		drill ( <i>Task</i> )
		push_clamp ( <i>Task</i> )
		move_down ( <i>Task</i> )
		move_up ( <i>Task</i> )
		release_clamp ( <i>Task</i> )
		push_sort ( <i>Task</i> )
		release_sort ( <i>Task</i> )
process_2 ( <i>Process</i> )		advance_position ( <i>Task</i> )
		push_sort ( <i>Task</i> )
		release_sort ( <i>Task</i> )

physical concept description of the assembly line: *System*, *Station* and *Component*. *System* class is used for assembly line instances, representing an entire production line. *Station* contains different stations of a system which represent an assembly operation (such as piston stuffing). *Component* includes elements that are used for performing station operations (such as clamping).

## 4.2 Adding required object properties in the ontology model

As described previously, some object properties are already asserted when models are imported. This is because relationships described in models are not modified during the import process, which is just an aggregation of graphs

into the generated graph dataset. Hence, graphs that are linking instances with properties are not affected.

However, the new links that relate elements of different domain models are inserted once the unique ontology is being created. In this case study, the properties *performsOperation* and *performsTask* are added and linked between corresponding PPR ontology instances. The IRI of these properties is same as of the PPR ontology. These added properties in the second step of the approach are represented as UML direct association. Table 1 presents the relationships between instances of the model after the definition of the PPR ontology object properties. The names of instances are based on the information available in [27].

The two products which are added to the model (i.e. *product\_1* and *product\_2*) are shown in Table 1. These products are manufactured through slightly different man-

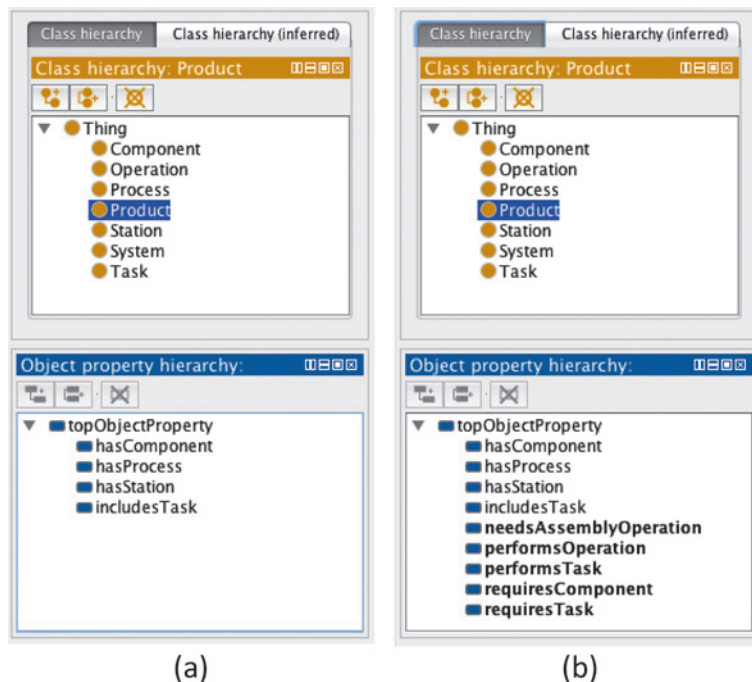


Figure 5: Class hierarchy and object property hierarchy (a) before and (b) after enriching the PPR ontology.

ufacturing processes. Thus, they are related to *op\_01* and *op\_02* that, in turn, are related to *process\_1* and *process\_2*. In this use case scenario, *product\_1* is executed using *process\_1*. The main steps constitute: advancing through the station, position testing, drilling and sorting. On the other hand, *product\_2* does not require the manufacturing operations of processing station and is transported to the next station through the positions of the station, which are the steps of the *process\_2*.

It should be noted that in addition to the shown object properties in Table 1, there are two more object properties represented in the UML ontology model: *requiresComponent* and *requiresTask*. These properties are used for determining which components and tasks are required to manufacture a product. However, they are represented within an UML derived association notation because such properties are derived from information included in the model. Basically, *requiresComponent* and *requiresTask* are not explicitly described because this approach implements such relation within semantic rules, as described in following sub-sections. Figure 5 shows the hierarchy of classes and object properties (a) before and (b) after the enrichment of object properties in the PPR ontology.

As it can be seen in Protégé views, the class hierarchy remains same in (a) and (b); the object property hierarchy changes because (b) contains the object properties shown in Figure 2 that are used for interrelating different domain models.

Figure 6 depicts that *processingStation* resource domain instance is linked to two domain instances. First, it

is still linked with *hasComponent* object property to other resource domain instances (from the imported resource ontology model) and, afterwards, it is linked with different process domain instances (*op\_01* and *op\_02*) with *performsOperation* object property.

### 4.3 Adding required SWRL rules and coupling data in the ontology model

In addition to presented classes and object properties that are imported from different domain models and object properties that are defined in the PPR ontology, the automatic coupling of data is achieved with SWRL rules. This section presents two rules that allow the mapping between product, component and tasks of assembly lines. As the rules are expressed through SWRL they are suitable for any RDF-based ontology model. Table 2 shows the two SWRL rules that are defined for this use case.

The first SWRL rule maps products with components by taking into account *performsOperation*, *includesTask*, *hasProcess*, *needsAssemblyOperation*, *hasComponent* and *performsTask* object properties and the classes that are linked to several model properties (i.e. *Component*, *Task*, *Operation*, *Station* and *Product*). The second SWRL rule maps products with tasks by taking into account *includesTask*, *hasProcess* and *needsAssemblyOperation* object properties and the classes that are linked to such properties (i.e. *Task*, *Process*, *Operation* and *Product*).

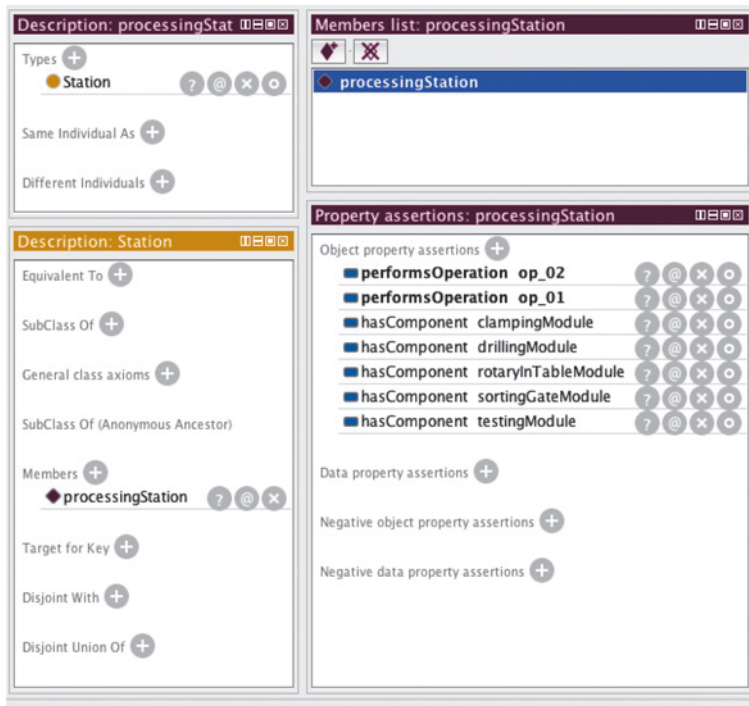


Figure 6: Object property assertions of *processingStation* in the PPR ontology.

Table 2: SWRL rules for automatic data coupling.

Rule 1: Linking products with required components	Rule 2: Linking products with required tasks
$\text{Component}(?c) \wedge \text{Task}(?t) \wedge \text{Operation}(?o) \wedge \text{Station}(?s) \wedge$ $\text{Product}(?pr) \wedge \text{performsOperation}(?s, ?o) \wedge$ $\text{includesTask}(?p, ?t) \wedge \text{hasProcess}(?o, ?p) \wedge$ $\text{needsAssemblyOperation}(?pr, ?o) \wedge \text{hasComponent}(?s, ?c) \wedge$ $\text{performsTask}(?c, ?t) \rightarrow \text{requiresComponent}(?pr, ?c)$	$\text{Task}(?t) \wedge \text{Process}(?p) \wedge \text{Operation}(?o) \wedge$ $\text{Product}(?pr) \wedge \text{includesTask}(?p, ?t) \wedge$ $\text{hasProcess}(?o, ?p) \wedge \text{needsAssemblyOperation}(?pr, ?o)$ $\rightarrow \text{requiresTask}(?pr, ?t)$

#### 4.4 Automatic data mapping within semantic reasoning engines

Using the presented SWRL rules, products can be inter-related with tasks and components. Thus, through SWRL rules and reasoning engines, the relation between products, tasks and components are inferred. It should be noted that these rules could be merged in one rule, which would give the same mappings. However, the implementation of two independent rules allows identifying which classes and properties are required for each type of mapping. It should be noted that for this approach implementation, Pellet reasoner [28] has been used as the reasoning engine for obtaining the mappings. Thus, before SWRL rules are executed, Pellet must be started.

Once the reasoning engine runs the final graph, dataset is generated because it includes the inferred data. Figure 7 shows the inferred data mappings of *product\_1* (visible in property assertions tab) created with the help of SWRL rules and the evaluation done by the semantic

reasoner. The mappings demonstrate that the data of all models is successfully coupled because *product\_1* (product domain) is linked to *op\_01* and to a set of tasks (process domain) and, at the same time, to a set of components (resource domain).

The presented implementation demonstrates successful coupling of models by (1) merging different domain models, (2) assertion of object properties between class instances and (3) automatic mappings through semantic reasoning, achieved through SWRL rules and model evaluation. If desired, the mappings of products and their requirements can be extracted from the model. As the model has been implemented in OWL (RDF-based) syntax SPARQL queries can be used for monitoring any information related to the model.

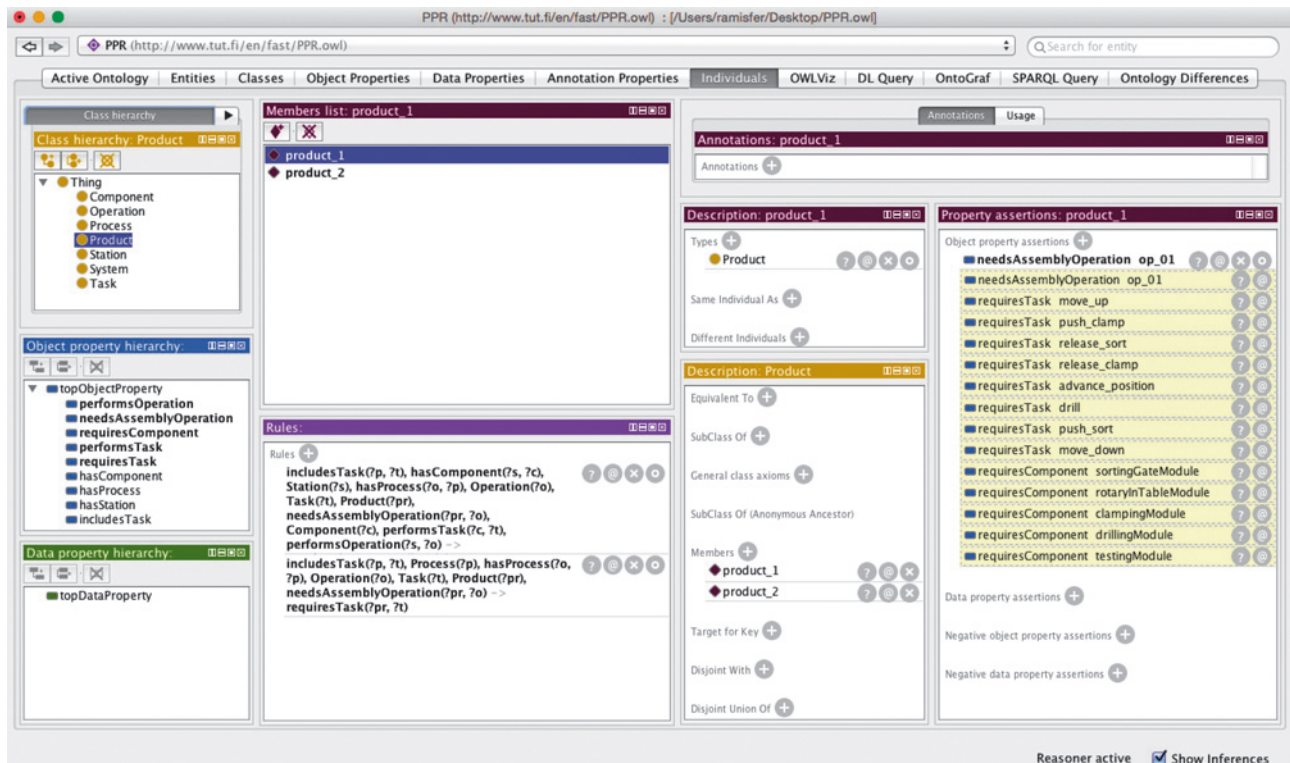


Figure 7: Inferred mappings in the PPR ontology.

## 5 Conclusion

This paper presents the application of a knowledge-based PPR mapping approach that can support dynamic configuration of assembly systems because the model interrelations may change when populating the ontology with new instances. Such approach relies on capturing and formalising in house engineering knowledge that link Product, Process and Resource in order to reduce system development and reconfiguration time. To illustrate the concept, generic models of products, processes and manufacturing resource components are created for a Festo test rig with two product variants scenario. The authors believe that such approach can be extended to allow reconfiguration of existing manufacturing systems in order to include more complex products (e.g. product structure and complexity, product variants) querying the available facilities to manufacture a new product variant. From the results of the case study, it can be concluded that ontological mapping of product, process and resource data and their representation within virtual modelling and simulation environment can, to a certain extent, allow the automation of the process that consists in instantiating a specific system from a library of PPR components and therefore to accelerate the configuration of manufacturing system for a specific production requirement.

The approach is implemented for a table-size test rig with a small number of components. Preliminary results are the automatic generation of a list of resource components based on Process and Product requirement input. Such results suggest that further development of ontology based system can potentially be used to achieve automatic configuration of manufacturing resources and contribute to the self-configuration paradigm introduced by Industry 4.0. An important aspect of the presented approach is the mapping between models based on query rules. The use of best practices to create these rules and serve as a reference to develop and advance the expertise is an important research area which needs to be addressed to enable wide application of such approach. A wider application of the proposed approach is required in order to assess the robustness, practicality and efficiency in terms of timesaving of this approach. Further research will focus on using semantics to characterise data collect from IoT devices deployed on the real system (e.g. PLC, energy monitors, sensors, production data) in order to further extend the data set based on which ontology and case based reasoning can be conducted.

**Acknowledgement:** The authors gratefully acknowledge the support of the UK EPSRC through the Knowledge-Driven Configurable Manufacturing (KDCM) research



project under the Flexible and Reconfigurable Manufacturing Initiative and the graduate school funding of Tampere University of Technology in carrying out this work.

## References

1. Kong, X., B. Ahmad, R. Harrison, Y. Park, and L.J. Lee. *Direct deployment of component-based automation systems*. In *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*. 2012.
2. Raza, M.B., *A knowledge based approach to integration of products, processes and reconfigurable automation resources*. 2012, Loughborough University.
3. Murray, S., *Knowledge management in manufacturing*, in *The Economist*. 2007.
4. Ferrer, B.R., B. Ahmad, A. Lobov, D.A. Vera, J.L.M. Lastra, and R. Harrison. *An approach for knowledge-driven product, process and resource mappings for assembly automation*. In *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*. 2015. IEEE.
5. Kong, X., *An approach to open virtual commissioning for component-based automation*, *Doctoral Thesis*, in Wolfson School of Mechanical and Manufacturing Engineering. 2013, Loughborough University: Loughborough, Leicestershire.
6. Falkman, P., F. Westman, and C. Modig. *Verification of operation sequences in process simulate by connecting a formal verification tool*. In *Control and Automation, 2009. ICCA 2009. IEEE International Conference on*. 2009.
7. Raza, M., B. and R. Harrison, *Design, development & implementation of ontological knowledge based system for automotive assembly lines*. *International Journal of Data Mining & Knowledge Management Process*, 2011. 1(5): p. 21–40.
8. Ferrer, B.R., B. Ahmad, A. Lobov, D. Vera, J.L.M. Lastra, and R. Harrison. *A knowledge-based solution for automatic mapping in component based automation systems*. In *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*. 2015. IEEE.
9. Haq, I., T. Masood, B. Ahmad, R. Harrison, B. Raza, and R. Monfared. *Product to process lifecycle management in assembly automation systems*. In *7th CIRP international conference on digital enterprise technology*. 2011. Athens, Greece.
10. Urrutia, U.A., P. Webb, and M. Summers. *Analysis of Design for X Methodologies for Complex Assembly Processes: A Literature Review*. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2014. American Society of Mechanical Engineers.
11. Choi, S.S., T.H. Yoon, and S.D. Noh, *XML-based neutral file and PLM integrator for PPR information exchange between heterogeneous PLM systems*. *International Journal of Computer Integrated Manufacturing*, 2010. 23(3): p. 216–228.
12. Brachman, R. and H. Levesque, *Knowledge representation and reasoning*. 2004: Elsevier.
13. Lastra, J.L.M., I.M. Delamer, and F. Ubis, *Domain ontologies for reasoning machines in factory automation*. 2010: ISA.
14. Bizer, C., T. Heath, and T. Berners-Lee, *Linked data-the story so far*. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 2009: p. 205–227.
15. Polleres, A., A. Hogan, R. Delbru, and J. Umbrich, *RDFS and OWL reasoning for linked data*, in *Reasoning Web. Semantic Technologies for Intelligent Data Access*. 2013, Springer. p. 91–149.
16. Kalibatiene, D. and O. Vasilecas, *Survey on ontology languages*, in *Perspectives in Business Informatics Research*. 2011, Springer. p. 124–141.
17. *OWL Web Ontology Language Semantics and Abstract Syntax*. 2015 [cited 2015 01 March]; Available from: <http://www.w3.org/TR/owl-semantics>.
18. *SPARQL Query Language for RDF*. 2015 [cited 2015 01 March]; Available from: <http://www.w3.org/TR/rdf-sparql-query>.
19. Ramis, B., L. Gonzalez, S. Iarovy, A. Lobov, J.L. Martinez Lastra, V. Vyatkin, and W. Dai. *Knowledge-based web service integration for industrial automation*. In *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*. 2014. IEEE.
20. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. [cited 2015 12 January]; Available from: <http://www.w3.org/Submission/SWRL/>.
21. Yap, C.E. and M.H. Kim. *Instance-based ontology matching with rough set features selection*. In *IT Convergence and Security (ICITCS), 2013 International Conference on*. 2013. IEEE.
22. Nath, R.P.D., H. Seddiqui, and M. Aono. *Resolving scalability issue to ontology instance matching in semantic web*. In *Computer and Information Technology (ICCIT), 2012 15th International Conference on*. 2012. IEEE.
23. Vegetti, M. and G. Henning, *ISA-88 Formalization. A Step Towards its Integration with the ISA-95 Standard*. FOMI'2014 Formal Ontologies meet Industry: p. 17.
24. Bosch, T. and B. Mathiak. *XSLT transformation generating OWL ontologies automatically based on XML Schemas*. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*. 2011. IEEE.
25. Belghiat, A. and M. Bourahla. *Transformation of UML models towards OWL ontologies*. In *Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference on*. 2012. IEEE.
26. Ensan, F. and W. Du, *A knowledge encapsulation approach to ontology modularization*. *Knowledge and information systems*, 2011. 26(2): p. 249–283.
27. Festo Didactic. *Processing station: Purely electrical*. 2015 [cited 2015 12 May]; Available from: <http://www.festo-didactic.com/int-en/learning-systems/mps-the-modular-production-system/stations/processing-station-purely-electrical.htm>.
28. Sirin, E., B. Parsia, B.C. Grau, A. Kalyanpur, and Y. Katz, *Pellet: A practical owl-dl reasoner*. *Web Semantics: science, services and agents on the World Wide Web*, 2007. 5(2): p. 51–53.

## Bionotes

**M. Sc. Borja Ramis Ferrer**

Tampere University of Technology – FAST-Lab. P.O. 600, FI-33101, Tampere, Finland

**borja.ramisferrer@tut.fi**

Borja Ramis Ferrer received the Ingeniero Técnico Industrial degree in electrical engineering from the Universidad de las Islas Baleares, Islas Baleares, Spain, in 2011 and the M.Sc. degree (with Distinction) in Factory Automation from Tampere University of Technology, Tampere, Finland, in 2013. He is currently working towards his Dr. Tech degree at Tampere University of Technology and is President's Doctoral School fellow. His research interests include the deployment of knowledge-based and cyber-physical systems in factory automation.

**Dr. Bilal Ahmad**

WMG, University of Warwick, Coventry, CV4 7AL, United Kingdom  
**b.ahmad@warwick.ac.uk**

Bilal Ahmad is a Research Fellow at WMG, University of Warwick. He received his MSc in Mechatronics and PhD in Automation Systems from Loughborough University. He specialises in the area of industrial automation. He has worked on a number of UK and EU engineering research projects in collaboration with automotive manufacturers, machine builders and control vendors to develop tools and methods to support lifecycle of automation systems.

**Dr. Daniel Vera**

WMG, University of Warwick, Coventry, CV4 7AL, United Kingdom  
**d.a.vera@warwick.ac.uk**

Daniel Vera has been working in the domain of manufacturing engineering for over ten years. His research interests are focused on various aspects of manufacturing from the modelling, analysis and optimisation of engineering processes to the design and development of 3D-based virtual engineering tools for supporting the manufacturing system lifecycle, which formed the focus of his PhD thesis. Dr Vera has been involved in numerous UK and European projects as a Research Associate at Loughborough University and now a Research Fellow at the University of Warwick. He is currently taking a leading role in the commercialisation of new automation systems and methods.

**Dr. Andrei Lobov**

Tampere University of Technology – FAST-Lab. P.O. 600, FI-33101, Tampere, Finland

**andrei.lobov@tut.fi**

Andrei Lobov is lecturing at the Tampere University of Technology. He received his PhD in Formal Methods of Factory Automation, in 2008. He holds BSc in Computer and System Engineering from the Tallinn University of Technology (2001). Then, he continued his education at the Tampere University of Technology and received MSc in Automation Engineering (2004). His research interests include development of architectures, methodologies and technologies for manufacturing systems. He is a technical coordinator of the eScop project.

**Prof. Dr. Robert Harrison**

WMG, University of Warwick, Coventry, CV4 7AL, United Kingdom  
**robert.harrison@warwick.ac.uk**

Robert Harrison is Professor of Automation Systems at WMG, University of Warwick and has been principal investigator on more than 35 EU, UK government, and commercial R&D projects related to manufacturing automation with current projects focusing on lifecycle engineering and virtual commissioning, control deployment and augmented reality. He led the UK research related to Ford's Technology Cycle Plan for powertrain manufacturing automation and was recipient of a RAEng Global Research Award to study "Lifecycle Engineering of Modular Reconfigurable Manufacturing Automation".

**Prof. Dr. José L. Martínez Lastra**

Tampere University of Technology – FAST-Lab. P.O. 600, FI-33101, Tampere, Finland

**jose.lastra@tut.fi**

José L. Martínez Lastra joined Tampere University of Technology in 1997, and became University Full Professor in 2006. His research interest is on applying Information and Communication Technologies to the fields of Factory Automation and Industrial Systems. Prof. Lastra leads the Factory Automation Systems and Technologies Laboratory with the ultimate goal of seamlessly integrating the knowledge of humans and machines. Prof. Lastra has co/authored over 250 scientific papers and holds a number of patents in the field of Industrial Informatics and Automation. He serves as Associate Editor of the IEEE Transactions on Industrial Informatics, and he is a Technical Editor of the IEEE/ASME Transactions on Mechatronics.